Briefing Document: Projects and Dissertations (The Pink Book)

Computer Science Tripos Part II

October 2018

Contents

1 Introduction

Candidates for Part II of the Computer Science Tripos are required to carry out a substantial piece of project work, and to submit a dissertation of about 10,000 words describing the project. The dissertation counts for about a quarter of the available marks in the Tripos.

In doing this, your objectives should be:

1. To display a range of Computer Science skills involved in the design, implementation and testing of a significant computer system. Usually this is a piece of software but it could be hardware or even the assembly of a knowledge base or a mechanically-assisted proof.

2. To demonstrate your ability to plan and carry out a large project in a coherent and effective way, adhering to the principles of design, quality and management required for good software engineering.

3. To show an understanding of the context in which your selected project lies. This includes the relationship of the task to the broad surrounding areas of Computer Science and other project-specific fields as well as an awareness of known results and the literature that supports your particular specialist area.

4. To select (and justify your selection of) suitable programming languages, techniques, algorithms, tools and data structures and convince the Examiners that you can learn new ones as necessary.

5. To plan and organise the collection and presentation of evidence that will show that the end result behaves in the way intended.

6. To prepare a formal report (the dissertation) in clear and concise expository form which will convince its readers that objectives 1–5 have all been achieved.

The project provides an opportunity to conduct a fairly detailed investigation of some area within Computer Science that particularly appeals to you. As long as the project meets the above formal criteria, you are free to suggest any project.

These notes are to give guidance about the selection, planning, execution and documentation of projects. They explain the arrangements that the Computer Laboratory makes to support and regulate project work, and comment about what the Examiners expect to find in dissertations. Since project work forms a substantial proportion of the year's work all of this is fairly important, and there is a lot to be said about it. It is suggested that this document be kept and occasionally checked throughout the year, particularly when the dissertation is being prepared, since otherwise it will be hard to keep track of all of the points that are made.

Important information on projects, including a hypertext version of this document, is available at:

`http://www.cl.cam.ac.uk/teaching/projects/`

2 Overview of the Project

Project planning and work goes on over a long period, and at different stages different aspects of the process come to the fore. At the start there is a briefing session where this document is issued and discussed. At the briefing session you will be allocated two Overseers, who are responsible for checking that your project is acceptable to the Laboratory.

You have a great deal of freedom in the selection of a project, and should start narrowing down the possibilities by identifying starting points or ideas that appeal to you. These initial ideas should be refined to a coherent project plan, which is then submitted as the project proposal. The proposal will be discussed informally with your Overseers, but is then submitted to the Head of the Computer Laboratory as a formal statement of intent.

Once the proposal has been accepted, work on the project can proceed. At about the halfway point the Laboratory will require you to present a short progress report to your Overseers.

In due course you write and submit a dissertation which will be about 10,000 words long. University Ordinances allow the Examiners to call students for a *viva voce* examination on the dissertation.

3 Timetable

This section indicates the critical dates and events throughout the year. These dates should be seen as immovable deadlines which must not be missed.[1] Enter them in your diary now!

3.1 Briefing session

A briefing session is given at the start of Full Term – see `http://www.timetable.cam.ac.uk` . Questions relating to this document will be answered. On the day of this session you will be learn who your Overseers are via the projects web page.[2]

3.2 Project Proposal

Over the first two weeks of term you submit three versions of your proposal, with the first two being drafts.

A 100-word outline of your project idea must be submitted to your Overseers on the Phase 1 Project Selection Status Report form by 3:00pm on Monday 8th October.

A draft project proposal must be submitted to your Overseers by 12 noon on Friday 12th October.

[1] See Section 14.

[2] `http://www.cl.cam.ac.uk/teaching/projects/`

The formal Project Proposal must be submitted to Student Administration in the Computer Laboratory by 12 noon on Friday 19th October. You should ensure that on delivery you are checked off the relevant list.

3.3 Progress Report Submission

Progress Reports (two paper copies) are due in the Student Administrator's office by 12 noon on Friday 1st February. Progress Report presentations and/or interviews with Overseers will take place over the following week or so.

3.4 Dissertation and Code Submission

The deadline for dissertation submission is **12 noon** on Friday 17th May. However, the Department strongly recommends you submit your dissertation a week earlier (i.e. the fourth Friday before exams start) so as to give you more time for revision.

Electronic submissions should be made using Moodle, in PDF form and the filename(s) should include your candidate number and the word 'dissertation' only. For the first time, from 2019, your work will be blind marked.

You will also submit, electronically, your source code and other relevant materials as a single ZIP file (or tar.gz file). This need not constitute a runnable demo of any sort, but must include all files you have authored or otherwise created (or modified in a substantial way) that you are claiming examination credit for.

Your submitted ZIP or .tar.gz file is subject to a 15 megabyte limit. The same limit applies to your dissertation PDF file(s), so the total you may upload is limited to 30 megabytes. If your work was substantial edits to a publicly available package (e.g. Linux kernel) that is much bigger, providing a patch file (diff -U) is one acceptable way of representing the work in a compact way.

Students can delete and reupload their files any time until the disseration deadline, so you are strongly advised to do a test upload early on.

In good time, Student Administration will notify all students about the procedure for submission of the PDF versions of the dissertation and the code files.

In addition, a project report form signed by the student's project Supervisor and Director of Studies should be completed on Moodle by 4pm on Wednesday 22nd May. The form can be found at

`http://https://www.vle.cam.ac.uk/login/index.php`

3.5 Viva Voce Examinations

The times of any CST *viva voce* examinations will be announced by 4pm on Friday 7th June and these examinations will probably take place on Friday 14th June.

Students *must* arrange to be available in Cambridge for *viva voce* examinations.

3.6 Dissertation Retention and Dissemination

The University will only use your submitted work for examination and accreditation purposes. It will not be disseminated or otherwise used by the University without your futher permission. The only exception is that a copy of your dissertation may be made available to University members who wish to look at it in the future.

You will be notified regarding the precise procedure for deleting your uploads after the publication of the class list.

4 Overseers

You will be allocated your Overseers on the day of the briefing session. If one of them subsequently becomes your project supervisor you will be changed to another overseeing group. Overseers are intended to provide impartial advice and are allocated so that nobody is simultaneously both an Overseer and Supervisor for any candidate. Overseers are available for (reasonable amounts of) discussion from the time of the briefing session up until the day on which proposals are submitted. During that critical planning period you have support both from two members of the Laboratory staff and from your College-organised Supervisor and Director of Studies.

Your proposal will include a brief description of the project **Starting Point** (see Section 7) or else state that you are starting from scratch.

When project proposals have been formulated, it is the Overseers who check them and recommend their acceptance to the Head of the Laboratory. Before submission, candidates must have talked to their Overseers about their ideas for projects and obtained informal acceptance of their plans based on near-final drafts of their proposals (see Section 6.8 for a detailed timetable). This ensures that the checking and formal approval processes will not cause trouble. It makes sense to give your Overseers the best possible chances of checking your plans early, and to take account of any issues that they raise. You will probably have most of your discussions with just one Overseer but you should send copies of draft proposals to both, since both will have to approve your final plan.

Your Overseers will need to be convinced that you understand the proposal, that it is a sound basis for a project without being too ambitious, that any special resources that will be required while carrying it out will be available, and that the proposal contains a suitably detailed work-plan with a timetable and list of milestones. When they accept a project they are agreeing that the description of it in the proposal is adequately detailed and that a reasonable candidate could complete a satisfactory piece of work given that specification. However, your Overseers will not have detailed knowledge of your particular strengths, weaknesses and background, so they are not in a position to certify that a project will be a great success for you in particular; of course they will be prepared to talk about such issues if you ask them.

The most efficient way to communicate with your Overseers is by e-mail. In this way you can send them drafts of your proposal and they can return comments much more quickly than by chasing each other around the Laboratory.

Overseers are not expected to invent projects, nor do they (in general) provide advice once

project proposals have been submitted. However, up until the proposal submission date they can provide useful advice and help.

5 Sources of Projects

The first stage in selecting a project is to collect a number of ideas. The main sources of inspiration are commonly:

1. Ideas proposed by candidates.
2. Suggestions made by Supervisors or Directors of Studies.
3. The project suggestions on the projects web page (see `http://www.cl.cam.ac.uk/teaching/projects/`).
4. Past years' projects: From 2017 onwards, the electronic copies of dissertations written by previous years' students are stored on the departmental filer. Any student wishing to gain access to these should ask the librarian. The titles of past projects can be searched at `http://www-dyn.cl.cam.ac.uk/~ncc25/projects.html`.
5. Proposals put forward by industry, especially companies who have provided vacation employment for students.

When ideas are first being suggested or discussed it is good to keep an open mind about them – a topic which initially seems very interesting may prove unreasonable on further consideration, perhaps because it will be too difficult. Equally, many of the ideas suggested by Laboratory members will relate to ideas that are unfamiliar to you, so will need study before you can appreciate what would be involved in following them. Almost all project suggestions should also be seen as starting points rather than fully worked out prescriptions. By making adjustments to original ideas, or selecting aspects of the project to concentrate on, even the most uninspired starting point can grow into a worthwhile proposal that has its own special character.

At an early stage it is usually best to identify one or two ideas that have the following properties:

1. Your Supervisor and Overseers agree that there could be an acceptable project based on the idea.
2. You can imagine being interested in work in the general area concerned.
3. You have identified somebody who is able and willing to supervise such a project.

Often (3) will solve itself first if you are picking up an idea proposed by a Supervisor or other member of the Laboratory.

You should bear in mind that the Examiners will require electronic submission of your dissertation and code. Therefore you should not sign anything, such as a non-disclosure agreement, that would prevent you from submitting them.

6 From Idea to Definite Plan

Turning a rough idea into a well thought out and clearly presented project plan can be a large amount of work. This section suggests some of the variety of things to keep in mind when planning. The amount of effort needed at this stage will also vary greatly from project to project. At one extreme will be ideas that come from a Supervisor, where major details have already been identified and which use only standard computing resources. At the other will be ones that start off nebulous, or with a clear ideal objective but no clear understanding of whether it can be achieved within the year. Throughout the planning phase the things that can help you most will be strong leads from project originators and the judgement of those (e.g. Overseers) who have seen the process unfold before.

6.1 First filtering

Some project ideas can be discarded very quickly as inappropriate. It is almost always best to abandon a doubtful idea early on rather than to struggle to find a slant that will allow the Overseers to accept it. Projects are expected to have a significant Computer Science content; for example, writing an application program or game-playing program where the main intellectual effort relates to the area supported rather than to the computation will not be suitable. Projects must also be about the right size to fit into the time available. The implications of this will best be judged by looking at past years' projects and by discussing plans with a Supervisor or Overseer. They should not allow you to waste much time considering either ideas that would prove too slight or ones that are grossly overambitious.

6.2 Resource availability

Each project will have a number of critical resources associated with its completion. If even one of these fails to materialise then it will not be possible to proceed with a project based on the idea; your Director of Studies can help you judge what might be a limiting issue.

In some cases a project may need to build on algorithms described in a technical report or other document known to exist but not immediately available in Cambridge. If a proper demonstration of a project will rely upon the availability of bulk data (e.g. a sample database, or machine-readable versions of the text of a novel) then this must be considered critical even if work could start without the data.

Non-standard hardware is probably the form of resource that can cause most trouble here, where the term "non-standard" is to be taken to mean anything other than a normal student account on Computing Service equipment (e.g. MCS). Thus other workstations (e.g. research machines in the Computer Laboratory), elaborate graphics support, private computers, projects involving the construction of hardware (including getting chips fabricated via the organisers of the ECAD course) must count as special. Similarly the use of software beyond that already installed by and supported by the proprietors of the selected machines cannot be automatically assumed. Further information on special resources is given at `http://www.cl.cam.ac.uk/teaching/projects/special.html`

It is reasonable to suppose that disc space and machine time will be made available in amounts adequate for all but extreme projects, but those whose ambitions lean

towards large databases or inherently lengthy calculations will have to check that they can be supported. In either case, a reasoned estimate of the resources required should appear in the project proposal. A modest increase in disc allocation on the MCS will be granted automatically, but you should read `http://www.cl.cam.ac.uk/teaching/projects/special.html` for further details and how to apply.

With these points in mind it is critical that the resources needed for any particular project be identified as early as possible – by project proposal time it will be necessary to have formal documented clearance to use any non-standard facilities.

The project proposal must contain as its last section a **Resources Declaration**. This must explicitly list the resources needed and give contact details for any person (apart from yourself) responsible for ensuring their availability. In particular, you should name the person responsible for you if your work requires access to the Department research area. The signatures of these people should also be present on the project cover sheet before submission.

If you are using your own computer, please state its specifications and also state your contingency plan in case it should fail (such as using MCS or another personal computer). Please also state your file back up plan and revision control system if used (which is recommended). If using your own computer please include the following text in your declaration: 'I accept full responsibility for this machine and I have made contingency plans to protect myself against hardware and/or software failure.'

6.3 Working with human participants

If your project involves collection of data via surveys, interviews or online, release of instrumented software, fieldwork, or experiments with human participants, such as usability trials or asking people to evaluate some aspect of your work, then you must seek approval by submitting a human participants request to the departmental Ethics Committee and record that you have done this by ticking the appropriate box on your coversheet. Further information is on the web page:

`http://www.cl.cam.ac.uk/teaching/projects/humansubjects.html`

6.4 Supervision

In some cases the most critical problem will be finding a suitable project Supervisor, somebody whom you will see regularly to report your progress and obtain guidance about project work throughout the year. This might be one of your main course Supervisors or a separate, specialist project Supervisor, but it should not be assumed that a person suggesting a project will be willing to supervise it. Supervisors have to be appointed by your Director of Studies, but in most cases it will be left up to you to identify somebody willing and able to take on the task. The Overseers will be interested only in seeing that someone competent has agreed to supervise the project, and that your Director of Studies is content with that arrangement.

6.5 Block plan of the project

Large projects have to be broken down into manageable chunks if they are to be completed. List the key components that will go to make up your final product. Credit is awarded specifically for showing a professional approach using any relevant software engineering methods at all stages of project design, development and testing. Plan an order in which you intend to implement the project components, arranging that both the list of tasks and the implementation order provide you with a sequence of points in the project where you can assess progress. Without a set of milestones it is difficult to pace your work so that the project as a whole gets completed on time.

When you have decomposed your entire project into sub-tasks you can try to identify which of these sub-tasks are going to be hard and which easy, and hence estimate the relative amounts of effort involved in each. These estimates, together with the known date when the dissertation must be submitted, should allow you to prepare a rough timetable for the work. The timetable should clearly make allowance for lecture loads, vacations, revision and writing your dissertation. Looking at the detail of such a plan can give you insight into the feasibility of the project.

It will also be necessary to make decisions about operating systems, programming languages, tools and libraries. In many cases there will be nothing to decide, in that the essence of the project forces issues. In the past projects have been carried out in C++, C#, FSharp/OCaml, Prolog, Reduce and Java. Other languages that are supported by the Computing Service or are in regular use by a research group within the Laboratory will usually be acceptable.

Working in assembly code usually limits productivity too severely for it to make sense for project work, and BASIC becomes unduly clumsy as programs reach the scale expected of the project. Uncommon languages or ones where the implementation is of unknown reliability are not ruled out, but must be treated with care and (if at all possible) fall-back arrangements must be made in case insuperable problems are encountered. It is expected that students will be prepared to learn a new language or operating system if that is a natural consequence of the project they select.

6.6 Planning for success

Projects are planned at the start of the year, and consequently it can be hard to predict the results of decisions that are made; thus any project proposal involves a degree of risk. Controlling and managing that risk is one of the skills involved in bringing a project to a successful conclusion. It is clear where to start: you should identify the main problem areas early and either allow extra margins of time for coping with them or plan the project so that there are alternative ways of solving key problems. A good example of this latter approach arises if a complete project requires a solution to a sub-problem X and a good solution to X would involve some complicated coding. Then a fall-back position where the project can be completed using a naive (possibly seriously inefficient, but nevertheless workable) solution to X can guard against the risk of you being unable to complete and debug the complicated code within the time limits.

As well as balancing your risks, you should also try to plan your work so that writing it up will be easy and will lead to a dissertation in which you can display breadth as well as

depth in your understanding. This often goes hand in hand with a project structure which is clearly split into sub-tasks, which is, of course, also what you wanted in order that your management of your work on the project could be effective.

A good dissertation will be built around a varied portfolio of code samples, example output, tables of results and other evidence of the project's successful completion. Planning this evidence right from the start and adjusting the project specification to make documenting it easier can save you a lot of agony later on.

6.7 Re-use of projects that have been attempted in the past

Projects are intended to give you a chance to display your abilities as a computer scientist. You are not required (or indeed expected) to conduct research or produce radically new results. It is thus perfectly proper to carry out a project that has been attempted before, and it is commonplace to have two students in the same year both basing their projects on the same original idea.

In such cases it is not proper to run a simple action replay of a previous piece of work. Fortunately all projects of the required scale provide considerable scope for different approaches; producing a new variation on an existing theme will not be hard. Furthermore the report produced at the end of a previous attempt at a project will often identify areas that led to unexpected difficulties, or opportunities for new developments – both these provide good scope for putting a fresh slant on the ideas involved.

6.8 Preparing the Project Proposal and consulting Overseers

From the briefing session until the final draft of your project proposal is ready you should keep in touch with both your Overseers, making sure that they know what state your planning is in and that they have had a chance to read and comment on your ideas. In most cases the best way of contacting Overseers will be using e-mail, and you should make a point of checking daily for messages that may have been sent to you. Overseers will generally be reluctant to turn down a project outright, but if you feel that yours sound particularly luke-warm about some particular idea or aspect of what you propose you would do well to think hard (and discuss the issues with your Supervisor) before proceeding. If Overseers declare a project plan to be unacceptable, or suggest that they will only accept subject to certain conditions, rapid rearrangement of plans may be called for.

Dealings with your Overseers divide into three phases between the briefing session and submitting your proposal. Most of the communications will be best arranged by e-mail, making sure to send copies to both Overseers.

6.8.1 Phase 1: Selecting a topic

You might already have thought of a suitable topic by the briefing meeting; if you have not, then you need to work quickly. Please pay careful attention to the points raised in the briefing lectures regarding selection of an appropriate topic. You must certainly choose something that has a defined and achievable success criterion. Note also that the marking

scheme explicitly mentions preparation and evaluation, so please select something that will require a corresponding initial research/study phase and a corresponding (preferably systematic) evaluation phase (see Section **??**).

You should complete a copy of the "Phase 1 Project Selection Status Report" (available at `http://www.cl.cam.ac.uk/teaching/projects/phase1.html`) and e-mail it, as a plain text message (not Word, HTML, PDF, PostScript etc.), to your Overseers for their approval.
[Deadline: Noon on the Monday after the briefing]

6.8.2 Phase 2: Filling out details

The details will include:

- Writing a description, running to a few hundred words.
- Devising a timetable, dividing the project into about 10 work packages each taking about a fortnight of your effort. The first couple of these might well be preparatory work and the last three writing your dissertation, with the practical work in the middle. These should have identifiable deliverables and deadlines leading to submission of your dissertation at the beginning of the Easter Term. You will probably write your progress report as part of the fifth work package.
- Determining special resources and checking their availability.
- Securing the services of a suitable Supervisor.

Send all this to your Overseers and ask them to check the details.
[Deadline: Noon on the Friday one week after the briefing]

For more advice as to the content of your proposal, see Section 7.

6.8.3 Phase 3: Final draft

In the light of your Overseers' comments, produce a final copy in the standard format. You now need to secure the signatures of your Supervisor and Director of Studies (in that order) and of the proprietor of any special resources that you need to use.

You do not secure signatures from your Overseers at this stage. Simply submit the proposal.
[Deadline: Noon on the Friday two weeks after the briefing]

Shortly after submission the Overseers will check your proposal again and, assuming that the foregoing steps have been followed carefully, all should be well and they will sign the proposal to signify formal acceptance. If the proposal is not acceptable you will be summoned for an interview.

7 Submission and Content of the Project Proposal

Completed project proposals must be delivered to the Student Administrator by noon on the relevant day. You should ensure that your name is checked off the Student Administrator's

list when your document is accepted.

The sheets of paper making up a proposal must be firmly attached together (stapled or in a simple binder). When planning your submission you should allow yourself adequate time for printing.

The Model Project Proposals[3] (which were originally written for Diploma students) conform to the required layout of all project proposals. These Model Project Proposals should be inspected and the style used should be followed closely. The remainder of this section draws attention to some details of the requirements.

A project proposal is expected to be about 1000 words long, and must be printed single- or double-sided on A4 paper, the sheets being neatly stapled together. It consists of the following:

1. A standard cover sheet – see `http://www.cl.cam.ac.uk/teaching/projects/ProposalForm.pdf`
2. The body of the proposal (see below).

When emailing drafts of your proposal to overseers, please make sure they contain all of the information required on the final hardcopy coversheet.

In the case of projects that are to rely on support from outside the University it will be necessary to procure a letter from the sponsors that confirms both that their equipment will remain available right up to the end of the course and that they understand that the results of work done by students cannot be viewed as secret or proprietary. An Overseer will then countersign the letter to record acceptance of these assurances.

The body of the proposal should incorporate:

1. An introduction and description of the work to be undertaken.
2. A statement of the **Starting Point**.
3. Description of the substance and structure of the project: key concepts, major work items, their relations and relative importance, data structures and algorithms.
4. A criterion which can later be used to determine whether the project has been a success.
5. Plan of work, specifying a timetable and milestones.
6. Resource Declaration.

This text will expand on the title quoted for your project by giving further explanation both of the background to the work you propose to do and of the objectives you expect to achieve. Quite often a project title will do little more than identify a broad area within which you will work: the accompanying description must elaborate on this, giving details of specific goals to be achieved and precise characterisations of the methods that will be used in the process. You should identify the main sub-tasks that make up your complete project and outline the

[3] See `http://www.cl.cam.ac.uk/teaching/projects/model-proposals.html`

algorithms or techniques to be adopted in completing them. A project description should give criteria that can be used at the end of the year to test whether you have achieved your goals, and should back this up by explaining what form of evidence to this effect you expect to be able to include in your dissertation. For example, this summary might take the form

> This project is to do A. Doing A requires the development of B. B will be tackled via C. C will be evaluated by D. The project will therefore consist of [e.g.] two main pieces of work, X and Y.

A statement of the **Starting Point** must be present to ensure that all candidates are judged on the same basis. It should record any significant bodies of code or other material that will form a basis for your project and which exist at project proposal time. Provided a proper declaration is made here, it is in order to build your final project on work you started perhaps even a year earlier, or to create parts of your programs by modifying existing ones written by somebody else. Clearly the larger the input to your project from such sources the more precise and detailed you will have to be in reporting just what baseline you will be starting from. The Examiners will want this section to be such that they can judge all candidates on the basis of that part of work done between project proposal time and the time when dissertations are submitted.

Similarly, a proposal must specify what it means for the project to be a success. It is unacceptable to say "I'll just keep writing code in this general area and what I deliver is what you get". It is advisable to choose a reasonably modest, but verifiable, success criterion which you *are as certain as possible* can be met; this means that your dissertation can claim your project not only satisfies the success criterion but potentially exceeds it. Projects which do not satisfy the success criterion are, as in real life, liable to be seen as failures to some extent.

Preparing a properly detailed work-plan can often seem the hardest part of completing a project proposal. This plan should show how the complete project is split into two- or three-week work packets, with these all being well enough specified that there will be a chance as the work progresses to evaluate how well targets have been met. Particular care should go into the selection of the milestones that will be reached just before the time that the progress report will become due. The timetable should make allowance for disruption to project work during the weeks immediately leading up to the written examinations, and should include dissertation preparation as well as programming time.

8 The Project

In formal terms work on your project can begin only when the Laboratory has accepted your proposal. In practice waiting to see the official note to this effect is unnecessary and you should start as soon as informal agreement has been reached with your Overseers. If you have a clear idea of what project you want to do and are confident that it will prove acceptable you can start even earlier, but remember that anything done that early should be reported in your project proposal.

8.1 Early days

Your project as a whole will be a large piece of work, normally much larger than any piece of programming you have been responsible for before. It is therefore inadvisable to jump straight into the middle of coding at the very start. If you will be working in a programming language that will be new to you then you should practise and learn it by writing small-scale code fragments (perhaps related to the code you will eventually want). If you will be using some specialist hardware (say a graphics display) or some large library or package you will need to show that you are in control by preparing demonstrations of your ability to drive it. Do not rush headlong into the production of large bodies of monolithic code. The first few weeks of your project will often involve you in a substantial amount of reading: studying manuals, searching libraries for copies of research papers or technical reports and checking the details of algorithms in textbooks. The larger a piece of work is the more important it becomes to have a clear plan as to how it will be executed, so you should probably try to add more detail to the work-plan prepared for your proposal.

8.2 Moving on

It will help both you and your Supervisor if, early in the project year, you can generate a steady stream of small-scale but visible achievements so that both of you can see clearly that work is underway and progress is being achieved. It is necessary to keep project work moving all the time despite the conflicting demands of lectures and supervision work, since it is easy to let days of inaction stretch into weeks. Furthermore it is remarkably easy to forget what was going on even in your own programs, and more than a couple of days' break in work can disrupt the flow of your ideas. By the end of the year it is expected that candidates will be self-reliant and in almost full command of their programs, but at the start this will generally not be so. When you find yourself in difficulty, and having made some reasonable effort to resolve things for yourself, you should seek assistance promptly – in many cases your Supervisor will be able to resolve your difficulties quickly and painlessly. As you work you should be testing both your ideas and your code all the time. This is easiest if your entire design has a clear modular structure. You should be prepared to write temporary bodies of code by way of scaffolding to support components that you want to test. If you put extra print statements into your programs so that you can trace their behaviour you should aim for an ideal whereby your trace output is in the form in which you would present a worked example of your algorithm; it should be sufficiently detailed to show all important internal working, but concise enough to be readable. Trace output consisting of tens or hundreds of pages of numbers amounts to an admission of defeat.

Your project plan will have given you some idea about the eventual size of your programs. It is almost certain that you will need to keep the final version of your code in the form of a set of files which get separately compiled and then linked together. Although some of your early experiments may be conducted using a compile-load-and-go mode of work, it will probably be useful to organise yourself for module-by-module recompilation fairly early. On Unix systems you will probably rely on the `make` utility. Whatever machine you are using you should find out how to arrange that large recompilations or potentially lengthy test runs are executed with low priority or at off-peak periods.

8.3 Evaluation Methods

Your project report must describe how the project has been evaluated, and your overseers are likely to discuss your plans for evaluation. Evaluation involves collecting evidence that tests or demonstrates how far the project has met its objectives. Different kinds of evidence are appropriate to different types of project, and you should discuss this with your supervisor and overseers.

Collecting, analysing and presenting evidence for the evaluation of your project requires specific activities, and time for these should be allowed in your project plan. Evaluation methods should be selected according to the objectives of the project, for example:

- Objectives involving engineering performance: experiments to collect quantitative data comparing a baseline your work, as well as other (existing) solutions.
- Objectives involving proof of correctness: mathematical proof will be involved.
- Objectives involving functional performance: systematic and reproducible testing procedures will be involved.
- Objectives involving user productivity: measures of task performance will be involved.
- Objectives involving aesthetic or affective outcomes: qualitative judgment data will be involved.

Evaluation methods are discussed in various parts of the Tripos curriculum. Philosophy of evaluation, including comparison of these different approaches, is explicitly taught in HCI and Software Engineering courses. Other courses address specific techniques, but the project will require you to compare and select from the available methods you have been taught, as appropriate to the project objectives and the advice of your supervisor.

8.4 Keeping notes

When a project is complete it can often be hard to look back and remember what aspects of it had seemed particularly uncertain at the start, and to trace all the problems that were overcome on the way to the successful completion. A project log-book can provide a great deal of help here. It can be a diary that will let you keep track of where time has gone, a place to make notes on examples you want to include in your dissertation and a reminder of why certain design decisions were made. Such a log can obviously prove its worth at the end of the year when the dissertation is being written, but it can be equally important earlier by giving you a clear view about the rate at which you have been able to make progress, and hence an indication as to how you should plan for the future. In keeping such a log it is useful to record failures, frustrations and dead ends as well as successes, since you may well wish to cite some of these to support the choices that you make.

Overall, as you start work you need to keep in view the final objective, which is the preparation of a dissertation in which you submit clear evidence that you carried out a significant piece of work in a coherent and well organised manner, making proper use of known results and demonstrating your ability to plan and complete such work within a predefined time scale.

9 Backing up Files

When working with several thousand lines of code over a period of months it becomes important to consider file backup and recoverability, and you should organise your work so that the inevitable mistakes can easily be undone.

Modern computer systems are remarkably reliable. Those administered by the University or Department (e.g. the MCS) will have their file systems carefully and regularly backed up as protection against any hardware failure.

The Department observes that maybe one or two of its students suffer a serious computer failure each year. So while this is not very likely to hit you, you need to be protected in case it does. You should institute a regular schedule for backing up project files, probably using all of USB memory sticks, the MCS and cloud services. Links to Computing Service information on making backups can be seen at `http://www.ucs.cam.ac.uk/arch/intro`.

In practice the biggest danger to your files is not hardware failure but clumsy editing or confusion about file-names; when tired it is painfully easy to delete an important file instead of the temporary one you intended to discard. There are also times when you may discover that a full week's work of heavy adjustment to your code was in fact misguided and that the best thing to do would be to restore your files to an earlier state. You should therefore arrange to make regular safe copies of your files, and preserve several generations of them. Using revision control, such as git or subversion is strongly recommended. The situation when this becomes most critical is when you are working under most pressure, which is of course when making backups feels most like a piece of bureaucracy that wastes your time!

10 Changes to the Original Plan

Once you have started on a project it is expected that you will follow through the plan as laid out in your proposal. Small adjustments to the emphasis that you put on different aspects of the work and refinements to the plan made as you go are of course always acceptable. However, in a very few cases, candidates want or need to make larger changes, and this section discusses that possibility. There are two classes of circumstance that might force you to have to abandon a project part way through and re-design it from the start or seek another. The first would be if (despite proper checking earlier on) some vital piece of hardware, software or data suddenly became unavailable and no alternative could be found. Cases of this sort should be very rare given the processes involved in getting the Project Resource Form signed, but natural or man-made disasters (lightning strikes, fires, floods, ...) do sometimes occur, and it is not always possible to recover from them rapidly enough to allow a one-year project to proceed undisturbed. The second case arises when a candidate finds that work is progressing much more slowly than originally predicted and that it is unrealistic to expect that the targets originally set will be attained.

In both of these cases there are three steps involved in getting the project back under control:

1. Identify as promptly as possible that there is a problem which could potentially grow into a serious one. Get in touch with your Supervisor and discuss the issue, trying to

see whether there is an easy way to side-step the problem. [Regular milestones let you spot work-rate problems.]

2. Try to get the difficulty resolved, setting a fixed date and a clearly stated way of knowing whether your problems are over. [e.g. “If the extra hardware is delivered by next Friday I will be able to catch up”.]

3. If step 2 does not correct your problems, seek further help from your Director of Studies as well as your Supervisor and, if your project will have to end up being significantly different from that described in your project proposal, get in touch with your Overseers or the Briefing Officer.

It should be obvious that problems are much easier to resolve if found early, and if discussed with your various advisers. Large changes of direction in a project are very strongly discouraged, and you should expect Supervisors, Directors of Studies and the Overseers to suggest ways of getting approximations to the original work done. These may include simulating unavailable equipment, concentrating more on a secure (if perhaps unexciting) aspect of a project or re-arranging your affairs by giving up other activities to make more time available for project work.

11 Progress Reports

About halfway through the project you have to report on the progress that you have made. There is a formal requirement for a written report of 300 to 500 words, which will go to your Overseers who will check it.

The Progress Reports and Presentations are mandatory. Any student unable to attend the Presentation with his or her own Overseers should arrange to join another group, and must inform both sets of Overseers, and the Student Administrator. Any student who cannot attend any of the sessions must present a letter of excuse from the Senior Tutor of their College.

Two copies of the Progress Report, in hardcopy form, are required and should be handed in to Student Administration.

The report should be printed on A4 paper and should contain:

- Your name and e-mail address.
- The title of your project.
- The name of your Supervisor.
- The name of your Director of Studies.
- The names of your Overseers.
- An indication of what work has been completed and how this relates to the timetable and work plan in the original proposal. The progress report should answer the following questions:

  - Is the project on schedule and if not, how many weeks behind (or ahead)?
  - What unexpected difficulties have arisen?
  - Briefly, what has been accomplished?

  It should be possible to understand the progress report independently of the original proposal, thus 'I have completed implementing the wombat module' rather than 'I have completed points 1 and 3 in the proposal but not point 2'.

In straightforward cases (entirely on schedule), one side of A4 could suffice. If the project is in difficulties, a new workplan should be included.

In addition, students must make an oral report on their progress. Overseers will arrange a meeting attended by all members of their overseeing group (typically 8 to 10 people), and each member of the group will describe progress made so far in a 5-minute presentation to the whole group. This oral report should be carefully rehearsed. Note that:

- The use of slides projected from a laptop is encouraged, but an overhead projector can be made available if prior notice is given.
- No more than four slides can usefully be described in 5 minutes.

It is quickest if one of the group connects their laptop to the data projector and this is used throughout the session with the other group members having already copied PDFs of their presentation to that laptop.

However, individual laptop users should note the following:

- The data projector connector is a standard D-sub 15-pin VGA connector (Macintosh users beware). Please also note that data projectors often cannot display high screen resolutions.
- If using a laptop, *before* attending the presentation session please ensure, e.g. by practising on a spare display monitor, that you
  - know how to change the screen resolution, and
  - can enable the external VGA output.

The written report and oral report provide a natural opportunity to consider adjustments to your original plan and schedule. In many cases these will be minor. In a few cases, the Overseers may feel that there is a need to discuss any special difficulties which have arisen in a more private setting. In such circumstances they will arrange to meet you individually. Such a meeting would be in addition to the overseeing group's oral report meeting. You may request an individual meeting yourself if you feel that it is necessary; this request should be put in writing at the end of your written report.

Overseers will write a formal report, of a few sentences, that will go via student admin to your Director of Studies and Supervisor.

12 The Dissertation

The dissertation should be written for a technically competent reader who is not necessarily familiar with the particular aspects of Computer Science involved.

The main body of the dissertation, running from the first page of the introduction until the last page of the conclusions, shall not exceed 40 pages nor exceed 12,000 words in length. Students should ensure the main body of their dissertation (pages 3 onwards) as well as any appendices do not contain direct personal identifiers (i.e. their name or their CRSID).

Better grades will arise from clarity and ease of reading, good pictures, clear explanation, minimal jargon and appropriate use of equations. Writing a dissertation requires planning and time. Try to allow three or four weeks for the task.

Once uploaded onto Moodle, your work will be checked by *Turnitin* for plagiarism.

Dissertation PDF files must be

- formatted for A4 paper;
- Typeset in 12-point font with a minimum of 2 cm margins;
- less than 15 megabytes length;
- (ideally) use embedded fonts.

Examiners and Assessors are permitted to judge your work only through study of your dissertation, although they will require your original source code to be available for them to refer to in cases where clarification is needed. You will be notified of the process by which you should upload your dissertation and code shortly before the deadline for the submission. Information about the process can be found at `http://www.cl.cam.ac.uk/teaching/projects/submission.html` but, as the system is currently changing, this is not guaranteed to be accurate outside of the Easter Term. Some hints on using the MCS for producing PDF files can be found at

`http://www.ucs.cam.ac.uk/desktop-services/ds-print/pdf.html`

To facilitate the assessment process, the Examiners require the top-level structure of the dissertation to be strictly as shown below.

	Cover Sheet
	Declaration of Originality
	Proforma
	Table of Contents
Chapter 1	Introduction
Chapter 2	Preparation
Chapter 3	Implementation
Chapter 4	Evaluation
Chapter 5	Conclusions
	Bibliography
	Appendices
	Index
	Project Proposal

It is not the intention of the Examiners to constrain writers too greatly. Although the layout of the Cover Sheet and the arrangement of the Proforma are tightly specified, the organisation and length of each of the five chapters are allowed to vary considerably from one dissertation to another.

Further details are given below, and at the end of this document there is a copy of the Guidelines issued to Assessors. The marking scheme is included. Study these Guidelines carefully.

Your dissertation PDF must be prefixed with the following pages that must come before your main title page. We provide Word and Latex users with blank forms for these pages.

12.1 The Cover Page

The single Cover Page contains

- Your Name, *in the extreme top right-hand corner*.
- The Title of your Dissertation.
- The Examination for which you are a candidate.
- Your College and the Year in which you are submitting the Dissertation.

12.2 Declaration of Originality

All dissertations must include an anti-plagiarism declaration immediately *after* the Proforma, preferably on the same page if there is room. The declaration must have exactly the following syntax:

> I, [Name] of [College], being a candidate for Part II of the Computer Science Tripos, hereby declare that this dissertation and the work described in it are my own work, unaided except as may be specified below, and that the dissertation does not contain material that has already been used to any substantial extent for a comparable purpose.
>
> Signed [signature]
>
> Date [date]

You will sign an electronic form of the same declaration as you upload your PDF, so you do not have to put a signature inside the PDF file.

The University drafted the wording, which is similar to that relating to dissertations in a wide range of subjects; thus the “unaided except as may be specified below” clause merits some explanation:

1. The clause does not require acknowledgement of the project supervision or informal conversations with peers.

2. The clause is believed to be about collaborative projects which are not now permitted in Computer Science. As such it is not relevant to Computer Science dissertations.

3. This clause aside and notwithstanding 1 and 2, candidates are required to draw attention, in the Implementation chapter, to the parts of the work which are not their own, in accordance with section 12.7 of this document. Other acknowledgements should be given wherever appropriate.

12.3 The Proforma Page

The single Proforma Page is a preface that immediately follows the Declaration of Originality. The Proforma must be arranged thus: (no direct direct personal identifiers should be used here or in the following main body of the dissertation.)

- Your candidate number.
- The Title of your Project.
- The Examination and Year.
- Word-count for the dissertation.
- Final line count: Number of lines written by the *student* in the final version of their software work.
- Project Originator.
- Project Supervisor.
- At most 100 words describing the original aims of the project.
- At most 100 words summarising the work completed.
- At most 100 words describing any special difficulties that you faced.
  (In most cases the special difficulties entry will say “None”.)

It is quite in order for the Proforma to point out how ambitious the original aims were and how the work completed represents the triumphant consequence of considerable effort against a background of unpredictable disasters. The substantiation of these claims will follow in the rest of the dissertation.

Student Administration will not accept a copy that does not include the relevant Cover, Proforma and declarations.

12.4 Table of contents

This should list the contents in some sensible way.

12.5 Introduction

The Introduction should explain the principal motivation for the project. Show how the work fits into the broad area of surrounding Computer Science and give a brief survey of previous related work. It should generally be unnecessary to quote at length from technical papers or textbooks. If a simple bibliographic reference is insufficient, consign any lengthy quotation to an appendix.

12.6 Preparation

Principally, this chapter should describe the work which was undertaken before code was written, hardware built or theories worked on. It should show how the project proposal was further refined and clarified, so that the Implementation stage could go smoothly rather than by trial and error.

Throughout this chapter and indeed the whole dissertation, it is essential to demonstrate that a proper professional approach was employed.

The nature of this chapter will vary greatly from one dissertation to another but, underlining the professional approach, this chapter will very likely include a section headed “Requirements Analysis” and incorporate other references to software engineering techniques.

The chapter will cite any new programming languages and systems which had to be learnt and will mention complicated theories or algorithms which required understanding.

It is essential to declare the **Starting Point** (see Section 7). This states any existing codebase or materials that your project builds on. The text here can commonly be identical to the text in your proposal, but it may enlarge on it or report variations. For instance, the true starting point may have turned out to be different from that declared in the proposal and such discrepancies must be explained.

12.7 Implementation

This chapter should describe what was actually produced: the programs which were written, the hardware which was built or the theory which was developed. Any design strategies that looked ahead to the testing stage might profitably be referred to (the professional approach again).

Descriptions of programs may include fragments of high-level code but large chunks of code are usually best left to appendices or omitted altogether. Analogous advice applies to circuit diagrams.

Draw attention to the parts of the work which are not your own. The Implementation Chapter should include a section labelled ”Repository Overview”. The repository overview should be around one page in length and should describe the high-level structure of the source code found in your source code Repository. It should describe whether the code was written from scratch or if it built on an existing project or tutorial. Making effective use of powerful tools and pre-existing code is often laudable, and will count to your credit if properly reported.

It should not be necessary to give a day-by-day account of the progress of the work but major milestones may sometimes be highlighted with advantage.

12.8 Evaluation

This is where Assessors will be looking for signs of success and for evidence of thorough and systematic evaluation as discussed in Section 8.3. Sample output, tables of timings and photographs of workstation screens, oscilloscope traces or circuit boards may be included. A graph that does not indicate confidence intervals will generally leave a professional scientist with a negative impression.

As with code, voluminous examples of sample output are usually best left to appendices or omitted altogether.

There are some obvious questions which this chapter will address. How many of the original goals were achieved? Were they proved to have been achieved? Did the program, hardware, or theory really work?

Assessors are well aware that large programs will very likely include some residual bugs. It should always be possible to demonstrate that a program works in simple cases and it is instructive to demonstrate how close it is to working in a really ambitious case.

12.9 Conclusions

This chapter is likely to be very short and it may well refer back to the Introduction. It might properly explain how you would have planned the project if starting again with the benefit of hindsight.

12.10 Bibliography

It is common, but not mandatory, to have a Bibliography.

12.11 Appendices

Assessors like to see some sample code or example circuit diagrams, and appendices are the sensible places to include such items. Accordingly, software and hardware projects should incorporate appropriate appendices. Note that the 12,000 word limit does not include material in the appendices, but only in extremely unusual circumstances may appendices exceed 10–15 pages – if you feel that such unusual circumstances might apply to you you should ask your Director of Studies and Supervisor to apply to the Chairman of Examiners. It is quite in order to have no appendices. Appendices should appear between the bibliography and the project proposal.

12.12 Index

An Index is optional.

12.13 Project Proposal

A copy of the original project proposal *must* be included at the very end of the dissertation.

12.14 Director of Studies and Supervisor's Report

A report from the student's project Supervisor and Director of Studies can be found at `https://www.vle.cam.ac.uk/course/index.php?categoryid=13811` This must be submitted to the Student Administrator, preferably at the same time as the dissertation, and in any case by 4pm on the following Wednesday.

13 Project Assessment

A copy of the Guidelines issued to Assessors is included at the end of this document. The Guidelines show the marking scheme which the Assessors are asked to follow and the score sheet that is completed for each candidate.

Each dissertation is marked as follows:

	Marks
Professional Practice & Presentation	14%
Introduction and Preparation	26%
Implementation	40%
Evaluation and Conclusions	20%

Every dissertation will be read by at least two of the internal examiners. A viva voce examination or additional assessment by an expert may also be considered. A proportion will also be read by an external examiner.

13.1 Professional Practice & Presentation

The assessors will determine whether you have taken a professional and ethical approach in your work. In particular, they will check that you have used appropriate methods and tools, understood software licenses, deployed appropriate review and evaluation techniques and been aware of the social and ethical impact of your work. You must demonstrate a structured design approach, including high-level design planning, design-for-test, consideration of human factors and systematic evaluation including confidence metrics within your evaluation. You should explain how you would show conformance with appropriate legislation, such as that for intellectual property, data protection, human subjects and software licenses such as those for open source. Show that you understand

the consequences of your project (or a more fully-formed variant of it) in terms of how it might affect commercial markets, contribute to society and/or the research community.

Regarding presentation, assessors primarily require the dissertation to be literate and tidy. It is not necessary to spend hours using an advanced graphics design package but it *is* necessary to write with correct grammar, in a clear and focused expository style using properly constructed sentences.

Strict adherence to the top-level arrangement described in Section 12 is regarded as part of the Presentation. Candidates who fail to put their names on the top right-hand corners of cover sheets, misunderstand the phrase “at most 100 words”, or omit the Proforma altogether, will lose marks for Presentation.

13.2 The Five chapters

Most of the marks are scored in the five chapters in the body of the dissertation.

Assessors recognise that the precise partitioning prescribed by the five chapter headings will sometimes prove too serious a constraint. A writer might, for example, feel that it is essential to discuss some aspects of the Implementation in earlier chapters. Assessors will credit Implementation marks ahead of time in such circumstances. It is unnecessary to repeat the discussion in order to earn the marks.

13.3 The Appendices

The dissertation should be fully comprehensible without reference to any appendix. Do not rely on content in an appendix providing any examination credit because appendices are not marked. But a consequence of following up a reference to an appendix may be an adjustment to the mark for a chapter in the main body of the dissertation.

13.4 Difficulty

No marks are explicitly awarded for difficulty. Assessors are well aware that some projects are more challenging than others and take this into account as they read the dissertation.

A trivial example might be the comparison of two projects which are very much the same except that one is written in Java and the other in BCPL. The project written in BCPL will be regarded as a little more challenging if only because there is a course on Java but none on BCPL. In consequence an Assessor might expect marginally more from the candidate who wrote in Java.

14 Late Submission

The penalty for late submission of the dissertation is *extremely severe*. The formula is:

$$penalty = \frac{10 + n}{40} \times mark$$

where n is the integer part of the number of days late.

This formula comes into play immediately after the noon deadline, when a quarter of the marks are lost.

If there should be substantial network or system failures within the University of Cambridge (including the Computer Laboratory, but not including Colleges), on the morning of the dissertation deadline day, Student Administration will accept manual submissions. Whether substantial failures have occurred, and what the alternate procedure is, will be declared by The Examiners for Part II by the noon on the deadline day. A manual submission will most likely be the handing in of a USB flash drive along with a paper copies of the Cover, Proforma and declaration.

15 Plagiarism and Fraud

Project work is conducted in your own time and obviously not under constant control and supervision. It is expected that work will be done fairly, and that the dissertation will be a proper report on the work performed. If you get unusually large amounts of assistance during the year, or use code written by somebody else you must report it. Results shown in your dissertation must have been produced by your programs and not concocted. Obviously both general and particular claims (including ones made implicitly rather than explicitly) must be true. Note that none of these points prevent you from obtaining assistance with your project – they just require that you present a sufficiently detailed explanation of how your results were achieved to allow the Assessors to assess the strengths of your contribution.

The University view fraud in examinations as a most serious offence, and all staff members involved in the assessment of dissertations are expected to watch for and report any anomalies which could indicate its presence.

You should read the Department's advice on the avoidance of plagiarism:

`http://www.cl.cam.ac.uk/teaching/exams/plagiarism.html`

In particular make sure that you give proper acknowledgement to the ideas and work of others, as suggested there.

You should be aware that the electronic copy of your dissertation be submitted to plagiarism-detection software for checking.

16 Intellectual Property Issues

In general, students here own all intellectual property they create, and this extends to the project and dissertation. A small number of students, however, sign away some or all of their IPR, either as a condition of a sponsorship agreement or as a condition of working on a project with externally-funded colleagues. (In the latter case you might wish to discuss this with your Director of Studies before deciding to working on such a project.)

Provided such IPR has not been signed away, students are welcome and even encouraged to exploit their work commercially. However, some points are worth noting:

- Material being submitted for a UK patent requires absence of prior public disclosure. *If you plan to patent something then either omit it from your dissertation or file the patent first – even examining (consider e.g. plagiarism-detection software) may represent a form of disclosure and examiners will not sign NDAs).* Moreover, it is usually unwise to divert energy from your project into patents; if you do come up with valuable software your primary IP protection is likely to be your *copyright* in it.
- Copyright arises automatically in both text and code that you write, and in images and video you take. You do not have to do anything for this to happen, but adding a copyright notice (to works you want to protect) can help resolve disputes later.
- A University project and a commercial product are valued according to very different metrics. *Spend your time working to get a good final project mark, and only then worry about the possibility of making money.*

17 Viva Voce Examinations

The Examiners will issue a notice indicating whom they are calling for *viva voce* examination: only a small proportion of candidates are involved, and in recent years these have spanned the entire range of ability, not just concentrating on obvious borderlines. If selected for a *viva voce* examination you will be asked to make a short presentation and discuss your project. You might like to take along a copy of your program and any useful output from it not included in your dissertation. The *viva voce* examination is concerned only with your project, not with other aspects of the Computer Science course.

18 Guidelines for Assessors

The following two pages show the Guidelines issued to Assessors of Part II dissertations. The marking scheme is included. Study these Guidelines carefully.

Guidelines for Assessors – Project Dissertations

Here is a dissertation for marking. The notes below should be read in conjunction with sections 12 and 13 of this year's Briefing Document ("Pink Book"). These sections give details of how the candidates have been asked to organise their dissertations and how these are to be assessed.

Candidates have been asked, in section 12, to structure their dissertations strictly as follows:

	Cover Sheet
	Proforma
	Declaration of Originality
	Table of Contents
Chapter 1	Introduction
Chapter 2	Preparation
Chapter 3	Implementation
Chapter 4	Evaluation
Chapter 5	Conclusions
	Bibliography
	Appendices
	Index
	Project Proposal

The marking scheme is described in section 13 and corresponds to a maximum of 100 marks being assigned as indicated in the following table:

	Marks
Professional Practice and Presentation	14
Introduction and Preparation	26
Implementation	40
Evaluation and Conclusions	20

Notes:

- Professional Practice and Presentation: Check that a structured design and evaluation approach has been followed, including consideration of evaluation accuracy, conformance with legislation and awareness of any commercial and societal impact. Check that the dissertation has the required structure and that the Cover Sheet, Proforma and Project Proposal are present and correct. Give credit for literacy and narrative quality but evidence of desk-top publishing skills should gain only marginal credit.
- Introduction and Preparation: consider how well the candidate understood the task and analysed it. Give credit for a good introduction to the technical background, a coherent discussion of the problems and sensible planning. Effort spent getting to grips with obscure documentation can be counted!
- Implementation: seek evidence of skill, clear thinking and common sense. Consider how much work was carried out and take into account how challenging this was.
- Evaluation and Conclusions: consider what was and what was not achieved. Give credit for a proper professional and repeatable approach to evaluation and for an interesting conclusion.

- Overall: No marks are explicitly assigned for difficulty but clearly challenging projects should be rewarded more generously than undemanding projects. Give credit for background work such as learning a new system, new algorithms or a new body of theory. Anything which is not part of ordinary course work is 'new' (for example BCPL is not now included in any lecture course). Projects need not break new ground nor be original in concept.

Assessment Form

Please fill in this form and *retain it*. Pass the dissertation on to whoever is due to read it next (or retain it if you are the last reader).

Candidate:

College:

Title:

Assessor:

Assessment: Please complete the following score sheet . . .

	Marks	
Prof. Practice and Presentation	_____	[max. 14]
Chapters 1 and 2 Introduction Preparation	_____	[max. 26]
Chapter 3 Implementation	_____	[max. 40]
Chapters 4 and 5 Evaluation Conclusions	_____	[max. 20]
Total	_____	[max. 100]

A *viva voce* examination is recommended? [Yes/No]

Additional assessment by an expert is recommended? [Yes/No]

Remarks: Please provide about 30 words of comment . . .